Peningkatan Keamanan Pesan Berbasis Android Menggunakan Algoritma Kriptografi RSA

Rahmat Sulaiman[1], Marina Vebu[2]
Dosen Teknik Informatika[1], Mahasiswa Teknik Informatika[2]
STMIK Atma Luhur[1,2]
Jl. Jend. Sudirman, Selindung Baru, Pangkalpinang
e-mail: rahmatsulaiman@atmaluhur.ac.id[1], 1411500166@mahasiswa.atmaluhur.ac.id[2]

***Abstrak* - Keamanan dan kerahasiaan sebuah data atau informasi merupakan hal yang sangat penting, baik dalam suatu organisasi seperti perusahaan, perguruan tinggi, maupun individual. Itu dikarenakan seringkali data atau informasi yang penting kadang tidak sampai ke tangan si penerima atau juga bisa sampai ke tangan si penerima tapi data yang diterima tersebut disadap terlebih dahulu tanpa sepengetahuan pengirim maupun penerima itu sendiri. Untuk mengatasi masalah yang ada, peneliti akan membuat suatu aplikasi peningkatan keamanan pesan berbasis android menggunakan algoritma kriptografi RSA yang akan di implementasikan untuk aplikasi pesan pada *smartphone* android. Aplikasi ini digunakan untuk mengirim dan menerima pesan pada *smartphone* berbasis android dengan mengamankan atau menyembunyikan pesan asli. Metode yang digunakan Adalah algoritma kriptografi RSA yang menggunakan dua kunci berbeda dalam melakukan enkripsi dan dekripsi yaitu kunci publik untuk enkripsi dan kunci privat untuk dekripsi. Algoritma kriptografi RSA ditambahkan dalam aplikasi ini guna meningkatkan keamanan pesan berbasis android dengan menerapkan algoritma RSA dalam proses mengenkripsikan pesan dengan menggunakan *key* yang berupa angka-angka yang telah ditentukan pengirim, dan mendekripsikan pesan yang dikirim menjadi pesan asli, sehingga pesan tersebut cukup aman dan tidak akan terbaca oleh pihak yang tidak mempunyai hak atas pesan tersebut.**

***Kata Kunci* - *Android*, Algoritma RSA, Kriptografi, Keamanan, Pesan**

I. PENDAHULUAN

Keamanan dan kerahasiaan sebuah data atau informasi merupakan hal yang sangat penting, baik dalam suatu organisasi seperti perusahaan, perguruan tinggi, maupun individual. Itu dikarenakan seringkali data atau informasi yang penting kadang tidak sampai ke tangan si penerima atau juga bisa sampai ke tangan si penerima tapi data yang diterima tersebut disadap terlebih dahulu tanpa sepengetahuan pengirim maupun penerima itu sendiri. Dan bisa saja data asli tersebut dirubah menjadi data yang tidak sesuai, sehingga dapat menjatuhkan pihak si pengirim. Padahal data sebenarnya tidak seperti itu. Kriptografi dan steganografi merupakan ilmu atau seni yang mempelajari pengamanan pesan atau data dan kerahasiaan pesan atau data. Dalam kriptografi, terdapat 2 proses utama, enkripsi dan dekripsi. Enkripsi adalah proses penyandian pesan asli atau *plainteks* menjadi *cipherteks* (teks tersandi). Sedangkan dekripsi adalah proses penyandian kembali *cipherteks* menjadi *plainteks* [1,2].

Salah satu usaha untuk mengamankan data diantaranya dengan menggunakan kriptografi. Algoritma kriptografi yang sering digunakan dalam proses pengamanan data yaitu algoritma RSA. Algoritma RSA adalah algoritma yang mudah untuk di implementasikan dan dimengerti [2,3].

.

II. LANDASAN TEORI

A. *Model Perangkat Lunak RAD (Rapid Application Development)*

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.

B. *Algoritma RSA*

Untuk mengamankan data, salah satu cara dapat diterapkan suatu algoritma kriptografi untuk melakukan enkripsi. Dengan enkripsi data tidak dapat terbaca karena teks asli atau plaintext telah diubah ke teks yang tak terbaca atau disebut chipertext. Ada banyak algoritma kriptografi yang dapat digunakan, berdasarkan sifat kuncinya dibagi menjadi dua yaitu simetris yang hanya memakai satu kunci rahasia dan asimetris (public key algorithm) yang memakai sepasang kunci publik dan kunci rahasia [3,4]. Pada penelitian ini algoritma kriptografi yang akan digunakan adalah algoritma kriptografi asimetris RSA yang ditemukan oleh Ron Rivest, Adi Shamir, dan

Leonard Adleman pada tahun 1978 dan RSA merupakan singkatan inisial dari nama mereka bertiga. RSA digunakan karena merupakan algoritma kriptografi asimetris yang paling sering digunakan pada saat ini dikarenakan kehandalannya. Panjang kunci dalam bit dapat diatur, dengan semakin panjang bit maka semakin sukar untuk dipecahkan karena sulitnya
memfaktorkan dua bilangan yang sangat besar tersebut, tetapi juga semakin lama pada proses dekripsinya [6].

Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman .Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci pribadi. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin [4,5].

Prosedur Membuat Pasangan Kunci

- Pilih dua buah bilangan prima sembarang, p dan q.
- Hitung $r = p \times q$. Sebaiknya $p \neq q$, sebab jika $p = q$ maka $r = p^2$sehingga p dapat diperoleh dengan menarik akar pangkat dua dari r.
- Hitung $f(r) = (p - 1)(q - 1)$.
- Pilih kunci publik, *PK*, yang relatif prima terhadap $f(r)$.
- Bangkitkan kunci rahasia dengan menggunakan persamaan (5), yaitu $SK \times PK \equiv 1 \pmod{f(r)}$.

C. Pengembangan Sistem Berorientasi Objek dengan UML

UML dibangun dari diagram-diagram yang dapat digunakan untuk memodelkan sebuah sistem dari berbagai sudut pandang waktu yang berbeda dalam daur hidup perangkat lunak sebuah sistem. OOSE dikembangkan oleh Ivar Jacobson adalah metode desain berorientasi objek yang melibatkan *use case*. *Use case* Merupakan skenario untuk memahami *requirement user* terhadap sistem menggambarkan interaksi antara *user* dengan sistem, menggambarkan tanggung jawab dan keluaran sistem pada pengguna dapat digambarkan dengan teks tanpa aliran kejadian, teks dengan aliran data, dan formal dengan *pseudo code*. Metode yang paling banyak digunakan adalah menggunakan UML (*Unified Modelling Language*) [7, 9].

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa – bahasa berorientasi objek seperti *C++, Java,* atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C [9].

III. METODOLOGI PENELITIAN

A. Algoritma RSA

Sandi RSA merupakan algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktorkan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yang mudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisaa digunakan dalam pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

Besaran-besaran yang digunakan pada algoritma RSA adalah:

- p dan q bilangan prima (rahasia)
- $n = p \times q$ (tidak rahasia)
- $\varphi(n) = (p - 1)(q - 1)$ (rahasia)
- e (kunci enkripsi / kunci publik) (tidak rahasia)
- d (kunci dekripsi / kunci privat) (rahasia)
- P (*plainteks*) (rahasia)
- C (*cipherteks*) (tidak rahasia)

Plain teks yang akan dienkripsi dengan RSA Coding merupakan angka-angka, sedangkan pesan yang dikirimkan bisaanya berbentuk teks atau tulisan. Sehingga dibutuhkan suatu kode yang sifatnya universal untuk mengubah pesan teks menjadi plain teks dalam bentuk bilangan. ASCII (*American Standard Code for Information Interchange*) atau Kode Standar Amerika untuk pertukaran informasi merupakan suatu standar internasional dalam kode

huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

Tabel 1. ASCII

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

B. Model Pengembangan Sistem

Dalam pembuatan aplikasi peningkatan keamanan pesan yang berbasis *android* ini penulis menggunakan model pengembangan *system* yaitu model RAD (*Rapid Application Development)*. Berikut adalah tahapan-tahapan dan penjelasan pengembangan perangkat lunak dengan menggunakan model RAD (*Rapid Application Development)* [7,8].

Model RAD memiliki 3 tahapan sebagai berikut :

1) *Rencana Kebutuhan (Requirement Planning)*
   Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasikan tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasikan syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan.
2) *RAD design workshop*
   Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai workshop. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukan representasi visual desain dan pola kerja kepada pengguna. Workshop desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama workshop desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.
3) *Implementasi (Implementation)*
   Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama workshop dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sisem baru atau bagian dari sistem ujucoba dan kemudian diperkenalkan kepada organisasi.

C. Metode Pengembangan Sistem

Dalam penelitian kali ini, penulis menggunakan salah satu Metode Pengembangan Sistem yaitu Model RAD (*Rapid Application Development*).

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.

D. Tools Pengembangan Sistem

Dalam penelitian ini penulis menggunakan *tools* pengembangan sistem UML (*Unified Modelling Language*) yang terdiri atas :

- *Use case* Diagram
  Use case diagram dalam penelitian ini digunakan untuk menggambarkan aktivitas apa saja yang bisa dilakukan oleh *user* dalam aplikasi peningkatan keamanan pesan ini.
- *Activity* Diagram
  Activity diagram dalam penelitian ini digunakan untuk menggambarkan alur kerja sistem dalam aplikasi peningkatan keamanan pesan ini.
- *Sequence* Diagram
  Sequence diagram dalam penelitian ini digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan menjelaskan hasil percobaan dan juga pembahasan dari penelitian ini,. Pada Penelitian ini algoritma yang digunakan adalah algoritma RSA, dimana Algoritma RSA merupakan algoritma asimetris yang menggunakan dua kunci berbeda untuk proses enkripsi dan dekripsi.

Perumusan Algoritma RSA

- Rumus Kunci Publik
  Kunci publik (e) adalah kunci yang didapat dari hasil pemfaktoran φ(n), yang relatif prima terhadap φ(n). Misal : φ(n) = (p-1)*(q-1) = (11-1)(13-1) = 120
  Pemfaktoran dari 120 adalah 2, 3, dan 5. Karena nilai (e) tidak boleh sama dengan nilai hasil pemfaktoran, maka yang relatif prima terhadap 120 adalah 7. Jadi, e = 7

- Rumus Kunci Private
  Setelah mendapatkan nilai dari kunci publik/kunci enkripsi (e), maka dapat ditentukan nilai kunci privat/kunci dekripsi (d) dengan menggunakan rumus:

$$(e \cdot d) \bmod \varphi = 1$$

atau

$$d = \frac{1 + (k*\varphi)}{e}$$

Pada rumus di atas nilai k adalah 1,2,3,...dst, sampai hasil nilai d adalah bilangan bulat.

- Rumus Enkripsi
  Setelah mendapatkan nilai kunci publik dan nilai privat, maka dapat dicari hasil enkripsi dengan rumus

$$Ci = Pi^e \bmod n$$

- Rumus Dekripsi
  Untuk mencari nilai dekripsi, maka dapat dicari dengan rumus

$$Pi = Ci^d \bmod n$$

Analisis proses merupakan tahap menganalisis proses enkripsi dan dekripsi. Berikut ini cara mencarinya :

- p = 11

  q = 67
- n = p * q

  = 11 * 67

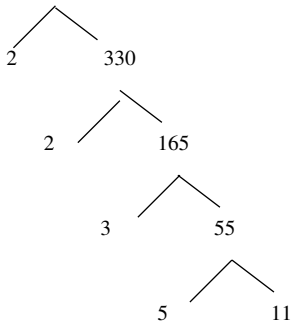
  = 737
- φ(n) = (p-1) (q-1)

  = (11-1) (67-1)

  = (10) (66)

  = 660
- FPB 660 = 2^2, 3, 5, 11 **jadi e = 13**

  660



- $d = \frac{1 + (k * \varphi)}{e}$

$$d = \frac{1 + (9 * 660)}{13} = \mathbf{457}$$

Plainteks = **M A R I N A**

K = **13**

Enkripsi = $Ci = Pi^e \bmod n$

C1 = M = $77^{13} \bmod 737 =$ **220**

C2 = A = $65^{13} \bmod 737 =$ **384**

C3 = R = $82^{13} \bmod 737 =$ **158**

C4 = I = $73^{13} \bmod 737 =$ **508**

C5 = N = $78^{13} \bmod 737 =$ **12**

C6 = A = $65^{13} \bmod 737 =$ **384**

Chipertext = **220 384 158 508 12 384 553 621 66 721**

Dekripsi = $Pi^d \bmod n$

P1 = $220^{457} \bmod 737 = 77 =$ **M**

P2 = $384^{457} \bmod 737 = 65 =$ **A**

P3 = $158^{457} \bmod 737 = 82 =$ **R**

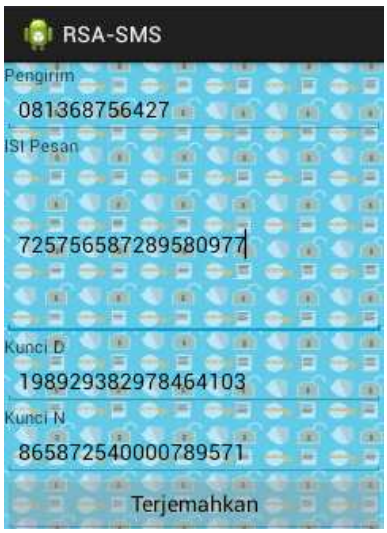
P4 = $508^{457} \bmod 737 = 73 =$ **I**

P5 = $12^{457} \bmod 737 = 78 =$ **N**

P6 = $384^{457} \bmod 737 = 65 =$ **A**

Gambar 1. Enkripsi dan Pengiriman Pesan Pada Android

Gambar 2. Hasil Enkripsi pada Android



Gambar 3. Isi Pesan yang diterima

Pada gambar diatas merupakan contoh aplikasi yang dibangun untuk menguji coba algoritma RSA yang sudah diterapkan untuk mengamankan pesan pada android, disitu terlihat bahwa kita ambil contoh pesan dengan angka-angka acak dan juga kunci seperti yang diatas. Pada gambar – gambar diatas nampak bahwa pesan yang dikirim otomatis terenkripsi, dan juga pesan diterima juga masih dalam terenkripsi, sehingga jika ingin membuka pesan tersebut harus menggunakan kunci yang sesuai.

V. PENUTUP

A. *Kesimpulan*

Keamanan pesan berbasis android dapat dibuat dengan menggunakan aplikasi *eclipse* dengan menambahkan algoritma kriptografi RSA. Dan menambahkan fitur enkripsi dan dekripsi dalam aplikasi ini. Fitur enkripsi digunakan oleh pengirim untuk mengirim pesan dalam bentuk pesan rahasia dengan menggunakan *key* yang telah ditentukan oleh pengirim, dan fitur dekripsi digunakan oleh penerima untuk membuka pesan rahasia dengan *key* yang sama dengan yang digunakan pengirim yang telah dienkrip dan dikirim oleh pengirim.Algoritma kriptografi RSA ditambahkan dalam aplikasi ini guna meningkatkan keamanan pesan berbasis android dengan menerapkan algoritma RSA dalam proses mengenkripsikan pesan dengan menggunakan *key* yang berupa angka-angka yang telah ditentukan pengirim, dan mendekripsikan pesan yang dikirim menjadi pesan asli, sehingga pesan tersebut cukup aman dan tidak akan terbaca oleh pihak yang tidak mempunyai hak atas pesan tersebut.

B. *Saran*

Dari hasil pembahasan di bab-bab sebelumnya, ada beberapa saran yang penulis berikan untuk mengembangkan sistem yang penulis buat, yakni sebagai berikut:

- Dapat dikembangkan dengan menggunakan algoritma dan metode yang lainnya, sehingga keamanannya lebih terjamin.
- Diharapkan kedepannya aplikasi ini tidak hanya dapat mengamankan pesan saja, tetapi dapat mengamankan seperti data pemerintahan, email dan media informasi lainnya.

DAFTAR PUSTAKA

[1] Sulaiman R, Isnanto B, 2018, “Peningkatan Keamanan Pesan Dengan Kriptogafi RC4 dan Steganografi LSB Pada File JPEG”, Konferensi Nasional Sistem Informasi 2018

[2] Syahputra, Edi, 2013, “Pengembangan Aplikasi Pertukaran SMS Rahasia Berbasis Android Menggunakan Algoritma RSA”, Institut Petanian Bogor (IPB), Bogor Agricultural University.

[3] Ginting, Albert, 2015, “Implementasi Algoritma Kriptografi RSA Untuk Enkripsi Dan Dekripsi Email”, Program Studi Sistem Komputer Fakultas Teknik Universitas Diponegoro, Jurnal Teknologi dan Sistem Komputer, Vol.3, No.2, April 2015, e-ISSN: 2338-0403.

[4] Fauzia, Rahma, Isna, 2015, “Rancang Bangun Aplikasi Enkripsi SMS (Short Message Service) Dengan Metode RSA Pada Telepon Selular Berbasis Android”, Politeknik Negeri Malang, Prosiding Seminar Informatika Aplikatif Polinema 2015 (SIAP), ISSN: 2460-1160.

[5] Nidya, Agustina, Ardelia, 2017, “Pengamanan Dokumen Menggunakan Metode RSA (Rivest Shamir Adleman) Berbasis Web”, Politeknik Negeri Sriwijaya Palembang, Prosiding Seminar Nasional Multi Disiplin Ilmu & Call For Papers UNISBANK ke-3 (SENDI_U 3) 2017 ISBN : 9-789-7936-499-93.

[6] Dewanto, I.Joko, 2013, “Pembuatan Aplikasi SMS Kriptografi RSA Dengan Android”, Universitas Esa Unggul, Forum Ilmiah Volume 10 Nomer 2, Mei 2013.

[7] Sutabri, Tata. 2012. Konsep Sistem Informasi. Yogyakarta: Andi Offset.

[8] Kendall, J.E. & Kendall, K.E. 2010. Analisis dan Perancangan Sistem. Jakarta: Indeks.

[9] Sulaiman R, Isnanto B, 2016, “Pembuatan Sistem untuk Peningkatan Kualitas Layanan Pada Lembaga Pendidikan Komputer XYZ” , Jurnal Teknologi Informatika dan Komputer Atma Luhur Vol 3. No 1. Maret 2016 ISSN: 2406-7962